Reasoning Under Uncertainty: Bnet Inference

(Variable elimination)

Computer Science cpsc322, Lecture 10

(Textbook Chpt 6.4)

June, 7, 2010

Lecture Overview

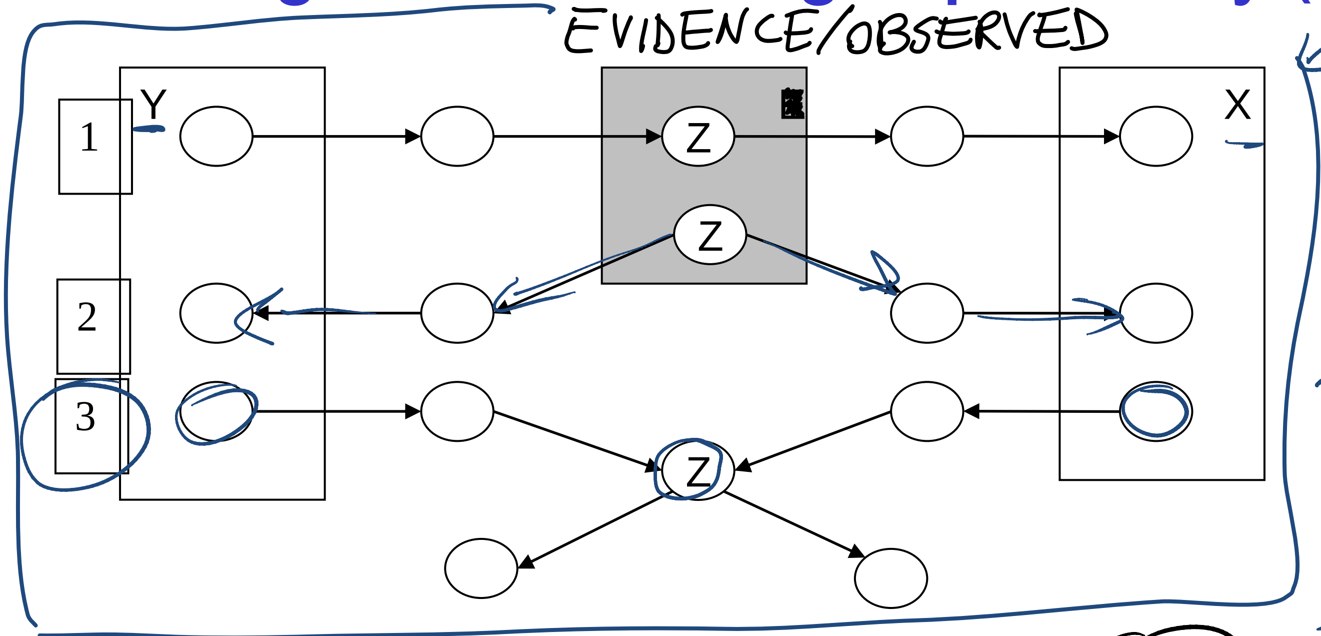
- **Recap Learning Goals last part of previous lecture**
- Bnets Inference
  - Intro
  - Factors
  - Variable elimination Algorithm
- Temporal Probabilistic Models

Learning Goals for last class

You can:

- In a Belief Net, determine whether one variable is independent of another variable, given a set of observations.
- Define and use **Noisy-OR** distributions. Explain assumptions and benefit.
- Implement and use a **naïve Bayesian classifier**. Explain assumptions and benefit.

3 Configuration blocking dependency (belief propagation)



true 2

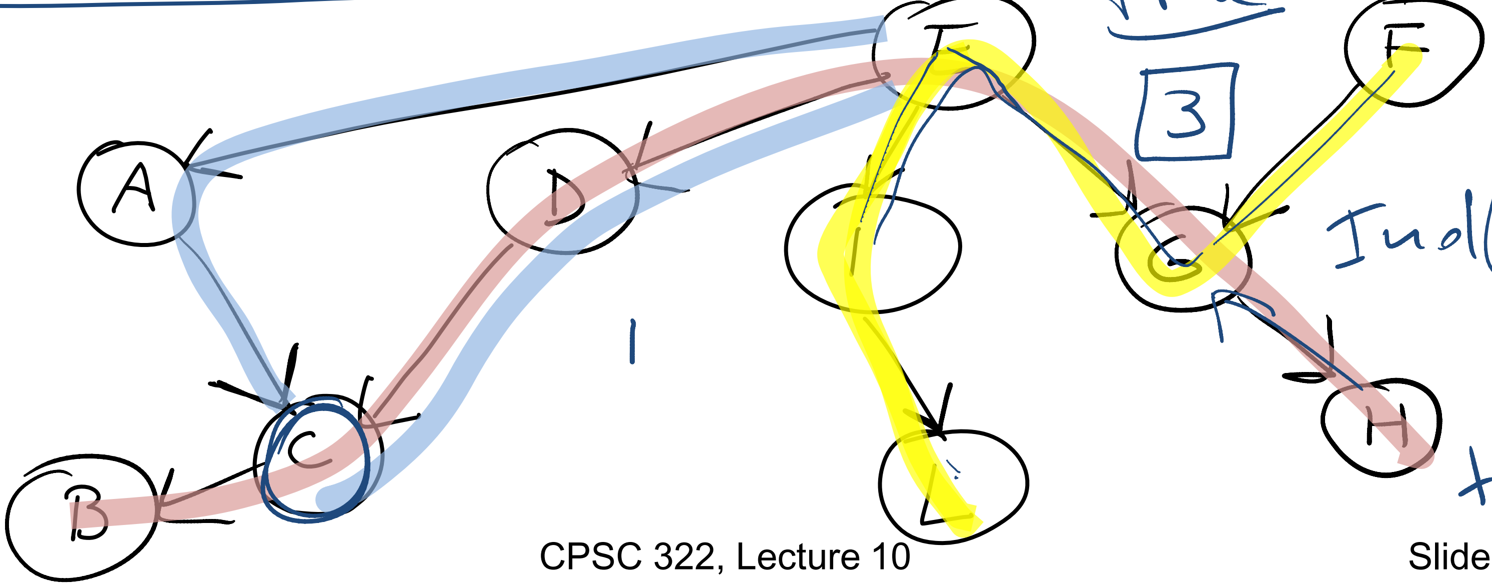
Indep(B,H {E})

false

Indep(E,C {D})

Indep(L,F {C})

true



Ind(L,F {H})

false

Naïve Bayesian Classifier

A very simple and successful Bnets that allow to classify entities in a set of classes C, given a set of attributes

Example:

- Determine whether an email is spam (only two classes spam=T and spam=F)
- Useful attributes of an email ? words contained in the email

Assumptions

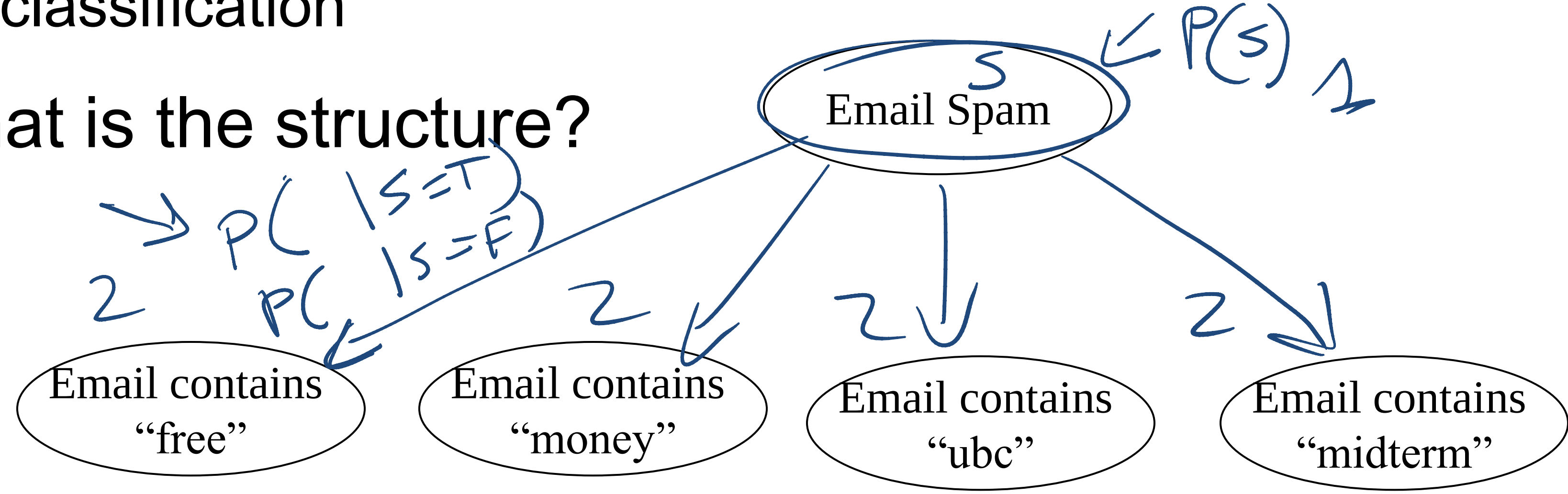
- The value of each attribute depends on the classification
- (Naïve) The attributes are independent of each other given the classification

P(“bank” | “account” , spam=T) = P(“bank” | spam=T)

Naïve Bayesian Classifier for Email Spam

Assumptions

- The value of each attribute depends on the classification ✓
- (Naïve) The attributes are independent of each other given the classification ✓
- What is the structure?



S

Email Spam

P(S) 1

$P(\mid S=T)$

$P(\mid S=F)$

2

Email contains “free”

2

Email contains “money”

2

Email contains “ubc”

2

Email contains “midterm”

words

10^5

in English

Number of parameters?

$2 \cdot 10^5$ vs 2^{10^5} Joint

Easy to acquire?

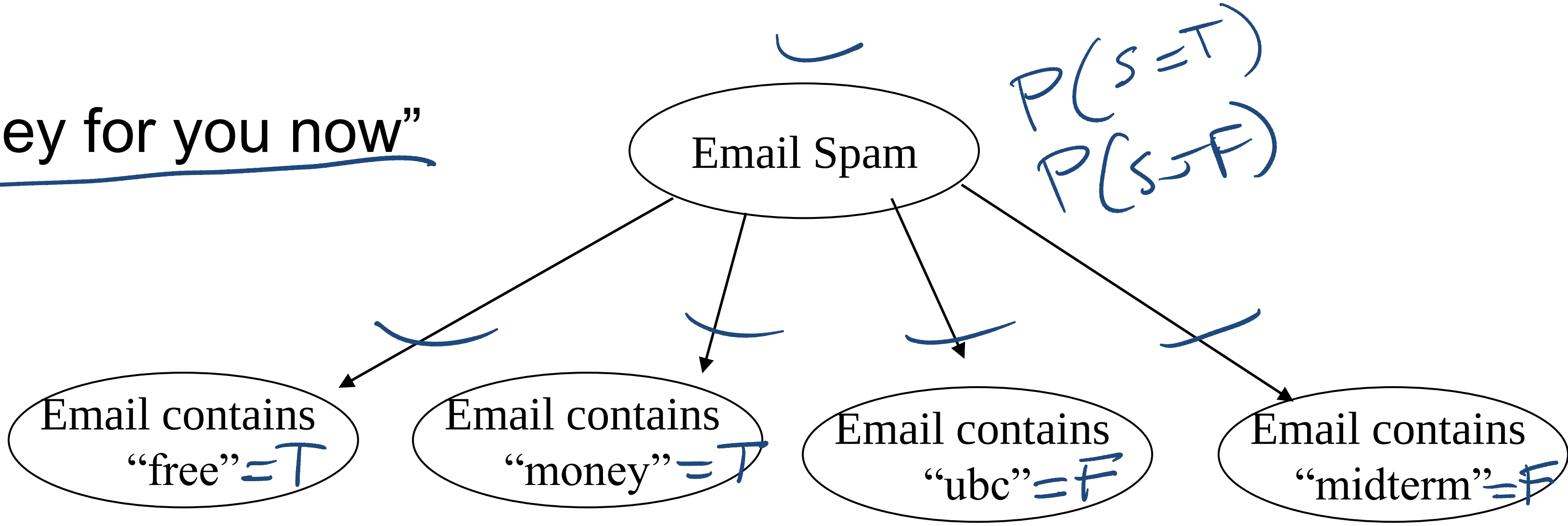
If you have a large collection of emails for which you know if they are spam or not……

NB Classifier for Email Spam: Usage

Most likely class given set of observations

Is a given Email *E* spam?

"free money for you now"



Email Spam

P(S=T)

P(S=F)

Email contains "free"=T

Email contains "money"=T

Email contains "ubc"=F

Email contains "midterm"=F

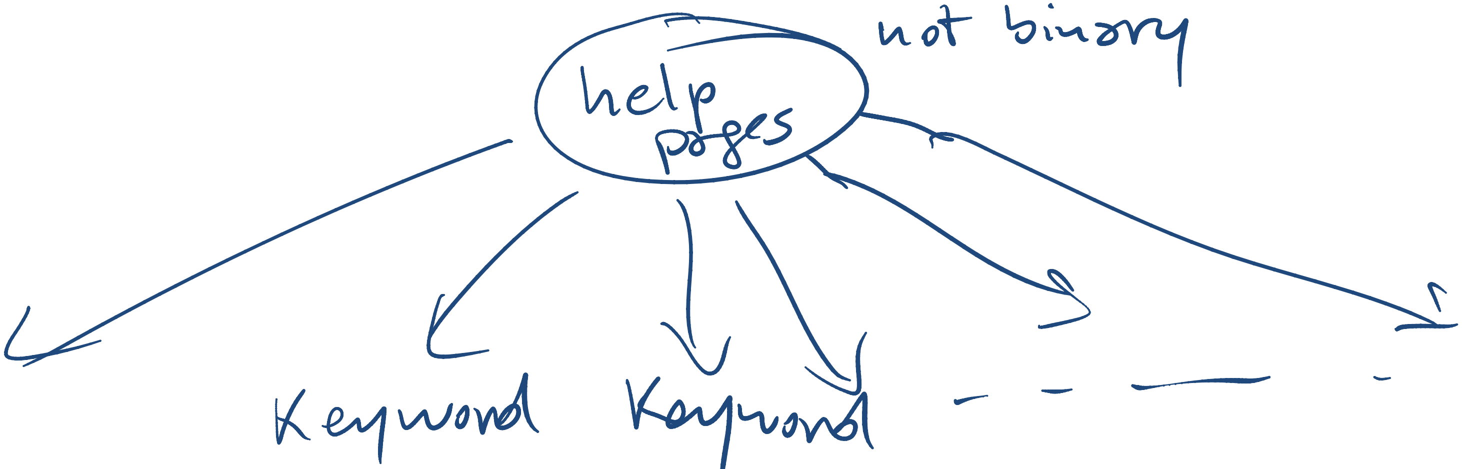
Email is a spam if……. $P(S=T) > P(S=F)$

after the two probs are updated in light of the evidence (words in email are set to T)

For another example of naïve Bayesian Classifier

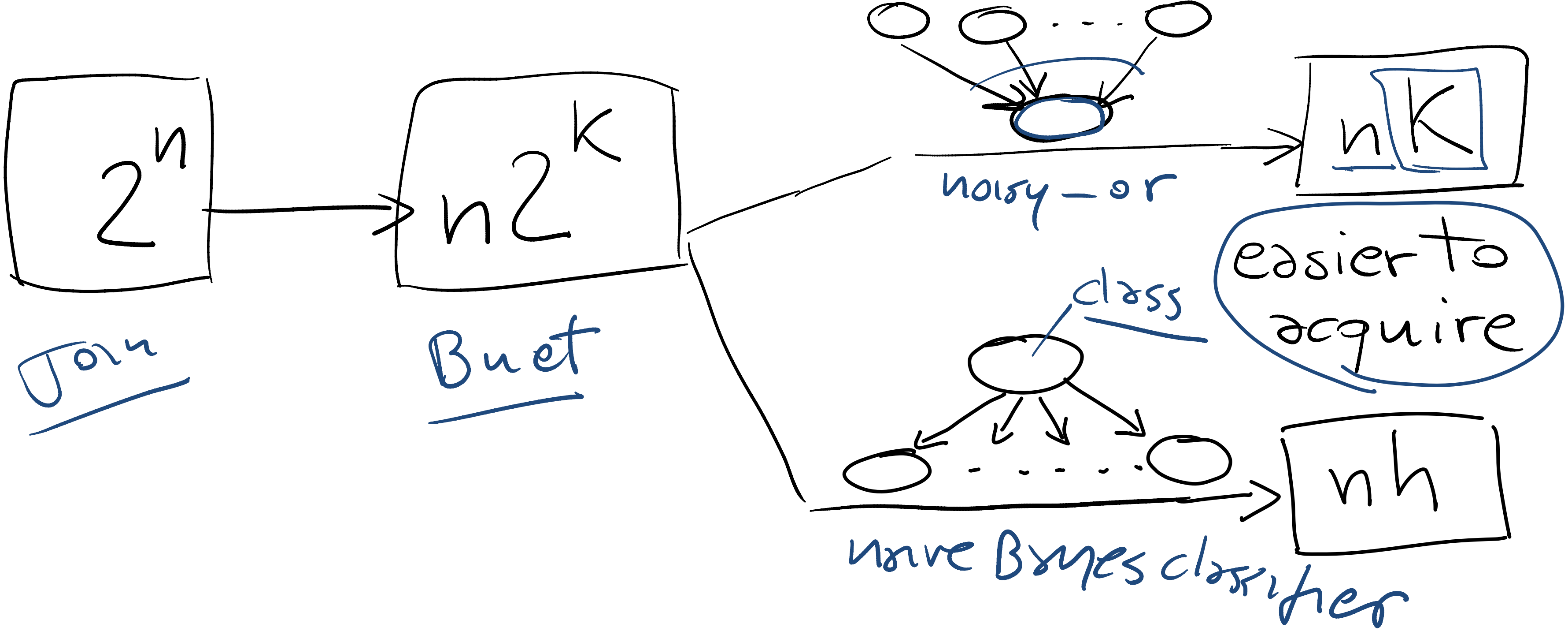
See textbook ex. 6.16

help system to determine what help page a user is interested in based on the keywords they give in a query to a help system.



Bnets: Compact Representations

n Boolean variables, k max. number of parents



Only one parent with h possible values

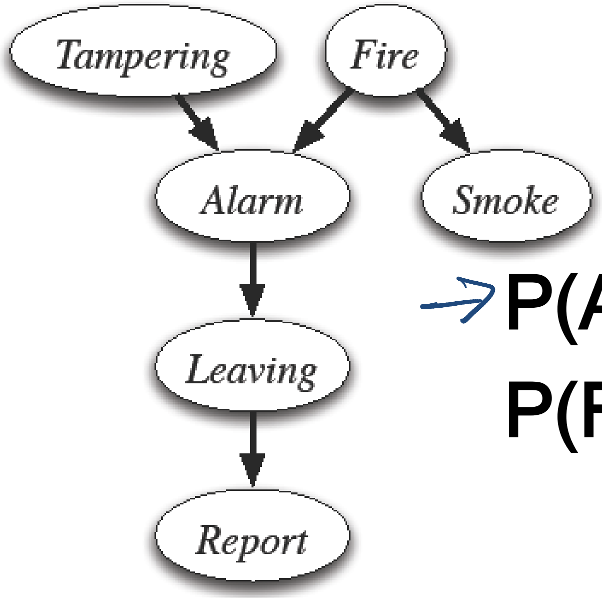
Lecture Overview

- Recap Learning Goals last part of previous lecture
- **Bnets Inference**
  - **Intro**
  - Factors
  - Variable elimination Algorithm
- Temporal Probabilistic Models

Bnet Inference

- **Our goal:** compute probabilities of variables in a belief network

  What is the posterior distribution over one or ~~more variables~~, conditioned on one or more observed variables?



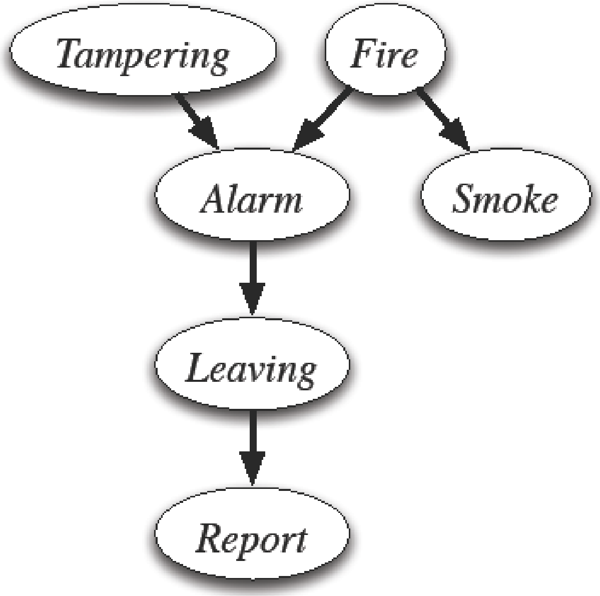
examples

→P(Alarm| Smoke = f)

P(Fire | Smoke = t ,Leaving = f)

Bnet Inference: General

- Suppose the variables of the belief network are $X_1,\ldots,X_n$.
- Z is the query variable
- *$Y_1=v_1, \ldots, Y_j=v_j$ are the observed variables (with their values)*
- $Z_1, \ldots, Z_k$ are the remaining variables
- What we want to compute: $\boxed{P(Z \mid Y_1 = v_1, \ldots, Y_j = v_j)}$



Example:

P(L | S = t , R = f)

$Z \Leftrightarrow L$

$Z_1 Z_2 Z_3 \Leftrightarrow T, F, A$

$Y_1 Y_2 \Leftrightarrow S, R$

What do we need to compute?

Remember conditioning and marginalization…

③

$$P(L \mid \underbrace{S = t , R = f}) = \frac{P(L, S=t, R=f)}{P(S=t, R=f)}$$

← ①

②

L	S	R	P(L, S=t, R=f)
t	t	f	,3
f	t	f	.2

Do they have to sum up to one?

no

② = .5

③

L	S	R	P(L | S=t, R=f)
t	t	f	.6
f	t	f	.4

In general…..

$$\underline{P(Z \mid Y_1 = v_1,\ldots,Y_j = v_j)} = \boxed{\frac{P(Z,Y_1 = v_1,\ldots,Y_j = v_j)}{P(Y_1 = v_1,\ldots,Y_j = v_j)}} = \frac{\overset{\text{①}}{\boxed{P(Z,Y_1 = v_1,\ldots,Y_j = v_j)}}}{\rightarrow \underset{\text{②}}{\sum_Z P(Z,Y_1 = v_1,\ldots,Y_j = v_j)}}$$

- We only need to **compute the** numerator and then **normalize**
- This can be framed in terms of **operations between $\boxed{\text{factors}}$** (that satisfy the semantics of probability)

Lecture Overview

- Recap Learning Goals last part of previous lecture
- **Bnets Inference**
  - Intro
  - **Factors**
  - Variable elimination Algorithm
- Temporal Probabilistic Models

Factors

- A **factor** is a representation of a function from a tuple of random variables into a number. [0, 1]
- We will write factor *f* on variables $X_1, \dots, X_j$ as $f(X_1 \dots X_j)$
- A factor denotes one or more (possibly partial) distributions over the given tuple of variables

  - e.g., $P(X_1, X_2)$ is a factor $f(X_1, X_2)$ *Distribution*
  - e.g., $P(X_1, X_2, X_3 = v_3)$ is a factor *Partial distribution* $f(X_1, X_2)_{X3 = v3}$
  - *e.g., $P(X \mid Z, Y)$ is a factor* *Set of Distributions* $f(X, Z, Y)$
  - e.g., $P(X_1, X_3 = v_3 \mid X_2)$ is a factor *Set of partial Distributions* $f(X_1, X_2)_{X3 = v3}$

P(Z|X,Y)

P(X,Y,Z)

P(Y|Z,X)

f(X,Y,Z)

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

Factors

- A **factor** is a representation of a function from a tuple of random variables into a number. $[0, 1]$
- We will write factor f on variables $X_1, \ldots, X_j$ as

$f(X_1 \ldots X_j)$

- A factor denotes one or more (possibly partial) distributions over the given tuple of variables

P(Z|X,Y)

- e.g., $P(X_1, X_2)$ is a factor $f(X_1, X_2)$ *Distribution*
- e.g., $P(X_1, X_2, X_3 = v_3)$ is a factor $f(X_1, X_2)_{X3 = v3}$ *Partial distribution*
- *e.g., $P(X \mid Z, Y)$ is a factor $f(X,Z,Y)$* *Set of Distributions*
- e.g., $P(X_1, X_3 = v_3 \mid X_2)$ is a factor $f(X_1, X_2)_{X3 = v3}$ *Set of partial Distributions*

f(X,Y,Z)

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

Manipulating Factors:

We can make new factors out of an existing factor

- Our first operation: we can *assign* some or all of the variables of a factor.

f(X,Y,Z):

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
~~f~~	~~t~~	~~t~~	~~0.4~~
~~f~~	~~t~~	~~f~~	~~0.6~~
~~f~~	~~f~~	~~t~~	~~0.3~~
~~f~~	~~f~~	~~f~~	~~0.7~~

What is the result of assigning X= t ?

f(X=t,Y,Z)

$f(X, Y, Z)_{X = t}$

More examples of assignment

r(X,Y,Z):

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

r(X=t,Y,Z):

Y	Z	val
~~t~~	~~t~~	~~0.1~~
t	f	0.9
~~f~~	~~t~~	~~0.2~~
f	f	0.8

r(X=t,Y,Z=f):

Y	val
~~t~~	~~.9~~
f	.8

r(X=t,Y=f,Z=f):

val
.8

Summing out a variable example

Our second operation: we can ***sum out*** a variable, say X_1 with domain $\{v_1, \dots, v_k\}$, from factor $f(X_1, \dots, X_j)$, resulting in a factor on $X_2, \dots, X_j$ defined by:

$f_3(A,B,C)$:

B	A	C	val
t	t	t	0.03
t	t	f	0.07
f	t	t	0.54
f	t	f	0.36
t	f	t	0.06
t	f	f	0.14
f	f	t	0.48
f	f	f	0.32

$\Sigma_B f_3(A,C)$:

A	C	val
t	t	.57
t	f	.43
f	t	
f	f	

$$\left(\sum_{X_1} f\right)(X_2,\dots,X_j) = f(X_1 = v_1, X_2, \dots, X_j) + \dots + f(X_1 = v_k, X_2, \dots, X_j)$$

Multiplying factors

- Our third operation: factors can be *multiplied* together.

$f_1(A,B)$:

	A	B	Val
* ●	t	t	0.1
○	t	f	0.9
	f	t	0.2
	f	f	0.8

$f_2(B,C)$:

	B	C	Val
●	t	t	0.3
*	t	f	0.7
○	f	t	0.6
	f	f	0.4

$f_1(A,B) \times f_2(B,C)$:

	A	B	C	val
●	t	t	t	.03
*	t	t	f	.07
○	t	f	t	.054
	t	f	f	
	f	t	t	
	f	t	f	
	f	f	t	
	f	f	f	

Multiplying factors: Formal

- The **product** of factor $f_1(A, B)$ and $f_2(B, C)$, where B is the variable in common, is the factor $(f_1 \times f_2)(A, B, C)$ defined by:

$$f_1(A,B)\, f_2(B,C) = (f_1 \times f_2)(A,B,C)$$

t f   t f

t f f   AB   BC

Note1: it's defined on all A, B, C triples, obtained by multiplying together the appropriate pair of entries from f_1 and f_2.

Note2: A, B, C can be sets of variables

Factors Summary

- A factor is a representation of a function from a tuple of random variables into a number.
  - $f(X_1, \ldots, X_j)$.
- We have defined three operations on factors:

1. Assigning one or more variables
   - $f(X_1=v_1, X_2, \ldots, X_j)$ is a factor on $X_2, \ldots, X_j$, also written as $f(X_1, \ldots, X_j)_{X_1=v_1}$

2. Summing out variables
   - $(\sum_{X_1} f)(X_2, \ldots, X_j) = f(X_1=v_1, X_2, ,X_j) + \ldots + f(X_1=v_k, X_2, ,X_j)$

3. Multiplying factors
   - $f_1(A, B)\ f_2(B, C) = (f_1 \times f_2)(A, B, C)$

Lecture Overview

- Recap Learning Goals last part of previous lecture
- **Bnets Inference**
  - Intro
  - Factors
  - **Variable elimination Algorithm**
- Temporal Probabilistic Models

Variable Elimination Intro

- Suppose the variables of the belief network are $X_1,\ldots,X_n$.
- Z is the query variable
- *$Y_1=v_1, \ldots, Y_j=v_j$ are the observed variables (with their values)*
- $Z_1, \ldots, Z_k$ are the remaining variables
- What we want to compute: $\boxed{P(Z \mid Y_1 = v_1, \ldots, Y_j = v_j)}$
- We showed before that what we actually need to compute is

$$\boxed{P(Z, Y_1 = v_1, \ldots, Y_j = v_j)}$$

This can be computed in terms of **operations between factors** (that satisfy the semantics of probability)

Variable Elimination Intro

- If we express the joint as a factor,

$$f(Z, \overbrace{Y_1,\dots,Y_j}^{\text{observed}}, \underbrace{\boxed{Z_1,\dots,Z_j}}_{\text{sum out}})$$

assign

- We can compute $\boxed{P(Z, Y_1=v_1, \dots, Y_j=v_j)}$ by ??
  - **assigning** $Y_1=v_1, \dots, Y_j=v_j$
  - and **summing out** the variables $Z_1, \dots, Z_k$

$$P(Z, Y_1 = v_1, \dots, Y_j = v_j) = \boxed{\sum_{Z_k} \cdots \sum_{Z_1}} \underbrace{f(Z, Y_1, .., Y_j, Z_1, .., Z_k)}_{\text{this is the Joint TOO BIG!}}{}_{Y_1=v_1,\dots,Y_j=v_j}$$

Are we done? no

Variable Elimination Intro (1)

$$\underline{P(Z, Y_1 = v_1, \ldots, Y_j = v_j)} = \boxed{\sum_{Z_k} \cdots \sum_{Z_1} \boxed{f(Z, Y_1, .., Y_j, Z_1, .., Z_k)}_{Y_1=v_1,\ldots,Y_j=v_j}}$$

- Using the chain rule and the definition of a Bnet, we can write $\underline{P(X_1, \ldots, X_n)}$ as $\prod_{i=1}^{n} P(X_i \mid pX_i)$

n

- We can express the joint factor as a product of factors

–

$f(Z, Y_1\ldots, Y_j, Z_1\ldots, Z_j)$ $\qquad \prod_{i=1}^{n} f(X_i, pX_i)$

✓

$$\underline{P(Z, Y_1 = v_1, \ldots, Y_j = v_j)} = \sum_{Z_k} \cdots \sum_{Z_1} \boxed{\prod_{i=1}^{n} f(X_i, pX_i)}_{\underline{Y_1=v_1,\ldots,Y_j=v_j}}$$

Variable Elimination Intro (2)

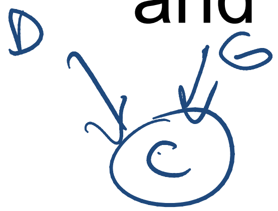
Inference in belief networks thus reduces to computing "the sums of products…."

$$\boxed{P(Z, Y_1 = v_1, \ldots, Y_j = v_j)} = \underbrace{\overset{\text{④}}{\sum_{Z_k}} \cdots \sum_{Z_1} \overset{\text{③}}{\prod_{i=1}^{n}} \overset{\text{①}}{\underline{f(X_i, pX_i)}} \quad \text{②} {}_{Y_1 = v_1, \ldots, Y_j = v_j}}$$

1. Construct a factor for each conditional probability.
2. In each factor assign the observed variables to their observed values.
3. Multiply the factors
4. For each of the other variables $Z_i \in \{Z_1, \ldots, Z_k\}$, sum out Z_i

How to simplify the Computation?

- Assume we have turned the CPTs into factors and performed the assignments



$$\boxed{f(X_i, pX_i)} \rightarrow \underline{f(\text{vars}X_i)}$$

$P(C|DG)$ $\underline{f(C,D,G)}_{\underline{G=t}} \rightarrow ?$ $f(C,D)$

$$\underline{\sum_{Z_k} \cdots \sum_{Z_1} \prod_{i=1}^{n} f(X_i, pX_i)_{Y_1=v_1,\ldots,Y_j=v_j}} \rightarrow \sum_{Z_k} \cdots \sum_{Z_1} \prod_{i=1}^{n} \underline{f(\text{vars}X_i)}$$

Let's focus on the basic case, for instance…

$$\boxed{\sum_A} \underline{f(C,D)} \times \underline{f(A,B,D)} \times \underline{f(E,A)} \times \underline{f(D)}$$

How to simplify: basic case

Let's focus on the basic case. $\sum_{Z_1} \prod_{i=1}^{n} f(\text{vars}X_i)$

$$\sum_{A} f(C,D) \times f(A,B,D) \times f(E,A) \times f(D)$$

- How can we compute efficiently?

Factor out those terms that don't involve $\boxed{Z_1}$!

$$\left(\prod_{\underline{i|Z_1 \notin \text{vars}Xi}} f(\text{vars}X_i) \right) \times \left(\sum_{Z_1} \prod_{\underline{i}|Z_1 \in \text{v}\underline{\text{ars}Xi}} \underline{f(\text{vars}X_i)} \right)$$

do not contain Z_1 do contain

$$f(C,D) \times f(D) \times \sum_{A} f(A,B,D) \times f(E,A)$$

General case: Summing out variables efficiently

$$\underline{\sum_{Z_k} \cdots \sum_{Z_1}} \underline{f_1 \times \cdots \times f_h} = \sum_{Z_k} \cdots \sum_{Z_2} (f_1 \times \cdots \times f_i) \left(\sum_{\underline{Z_1}} \underline{f_{i+1} \times \cdots \times f_h} \right)$$

$$\sum_{Z_k} \cdots \sum_{Z_2} \underline{f_1 \times \cdots \times f_i \times \boxed{f'}}$$

Now to sum out a variable $\boxed{Z_2}$ from a product $\underline{f_1 \times \ldots \times f_i \times f'}$ of factors, again partition the factors into two sets

- $\hat{F}$: those that contain Z_2
- F: those that do not

$$\prod F \times \sum_{Z_2} \prod \hat{F}$$

Lecture Overview

- Recap Learning Goals last part of previous lecture
- **Bnets Inference**
    - Intro
    - Factors
    - **Variable elimination Algorithm**
- Temporal Probabilistic Models

30 MINS BREAK

Analogy with "Computing sums of products"

This simplification is similar to what you can do in basic algebra with *multiplication* and *addition*

- It takes *14* multiplications or additions to evaluate the expression

$$\underline{a\,b + a\,c + a\,d + a\,e\,h + a\,f\,h + a\,g\,h.}$$

- This expression be evaluated more efficiently….

$$a*(b + c + d. + h*(e + f + g)) \quad 7 \text{ operations}$$

Variable elimination ordering

Is there only one way to simplify?

$$P(G,D=t) = \boxed{\sum_{A,B,C,} f(A,G)\ f(B,A)\ f(C,G)\ f(B,C)}$$

CBA

$$P(G,D=t) = \sum_{A} f(A,G) \sum_{B} f(B,A) \sum_{C} f(C,G)\ f(B,C)$$

BCA

$$P(G,D=t) = \sum_{A} f(A,G) \sum_{C} f(C,G) \sum_{B} \boxed{f(B,C)\ f(B,A)}$$

Variable elimination algorithm: Summary

$$P(Z, | \cancel{Y_1,\dots,Y_j}, \cancel{Z_1,\dots,Z_j})$$

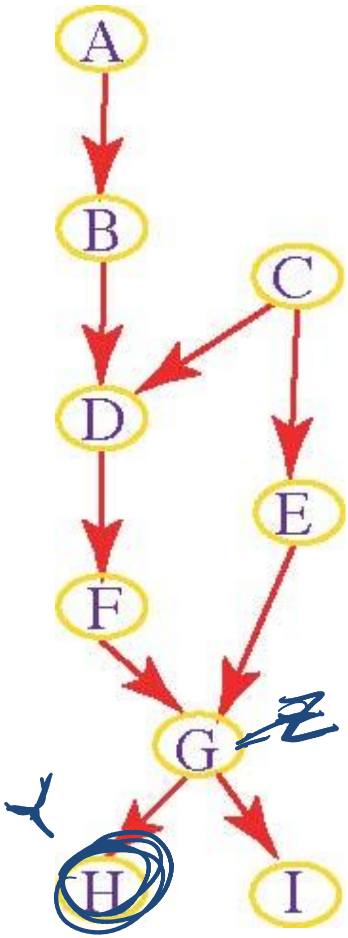
To compute $P(Z| Y_1=v_1 ,\dots ,Y_j=v_j)$:

1. Construct a factor for each conditional probability.
2. Set the observed variables to their observed values.
3. Given an elimination ordering, simplify/decompose sum of products
4. Perform products and sum out Z_i $Y_1=v_1$ Z
5. Multiply the remaining factors (all in ? Y_2 Z_2)
6. Normalize: divide the resulting factor $f(Z)$ by $\sum_Z f(Z)$.

Variable elimination example

Compute $P(G \mid H=h_1)$.

- $P(G,H) = \sum_{A,B,C,D,E,F,I} P(A,B,C,D,E,F,G,H,I)$



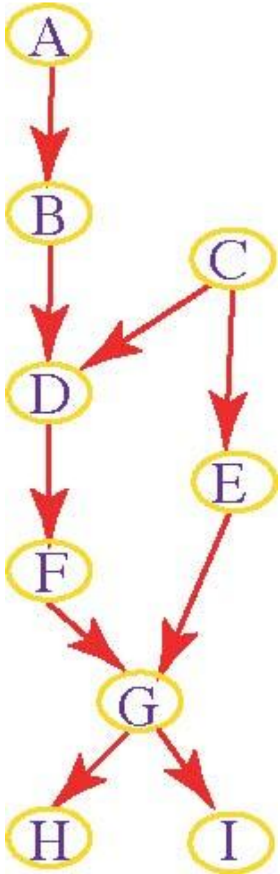
Variable elimination example

Compute $P(G \mid H=h_1)$.

- $P(G,H) = \sum_{A,B,C,D,E,F,I} \underline{P(A,B,C,D,E,F,G,H,I)}$

Chain Rule + Conditional Independence:

$$P(G,H) = \sum_{A,B,C,D,E,F,I} \underline{P(A)P(B|A)P(C)P(D|B,C)P(E|C)P(F|D)P(G|F,E)P(H|G)P(I|G)}$$



Variable elimination example (step1)

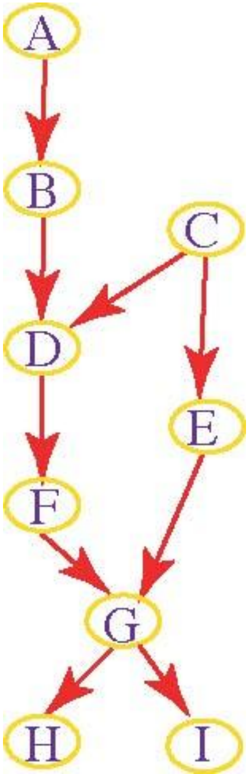
Compute $P(G \mid H=h_1)$.

- $P(G,H) = \sum_{A,B,C,D,E,F,I} P(A)P(B|A)P(C)P(D|B,C)P(E|C)P(F|D)P(G|F,E)P(H|G)P(I|G)$

Factorized Representation:

$$P(G,H) = \sum_{A,B,C,D,E,F,I} f_0(A)\, f_1(B,A)\, f_2(C)\, f_3(D,B,C)\, f_4(E,C)\, f_5(F, D)\, f_6(G,F,E)\, f_7(H,G)\, f_8(I,G)$$

- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$



Variable elimination example (step 2)

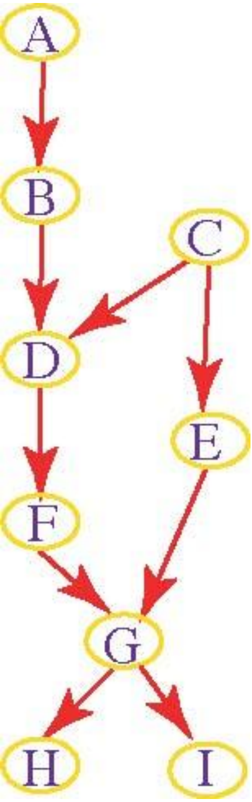
Compute $P(G \mid H=\boxed{h_1})$.

Previous state:

$$P(G,H) = \sum_{A,B,C,D,E,F,I} f_0(A)\, f_1(B,A)\, f_2(C)\, f_3(D,B,C)\, f_4(E,C)\, f_5(F,D)\, f_6(G,F,E)\, \underline{f_7(H,G)}\, f_8(I,G)$$

Observe H :

$$\underline{P(G,H=h_1) = \sum_{A,B,C,D,E,F,I} \boxed{f_0(A)\, f_1(B,A)\, f_2(C)\, f_3(D,B,C)\, f_4(E,C)\, f_5(F,D)\, f_6(G,F,E)\, \underline{f_9(G)}\, f_8(I,G)}}$$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

- $\underline{f_9(G)}$

Variable elimination example (steps 3-4)

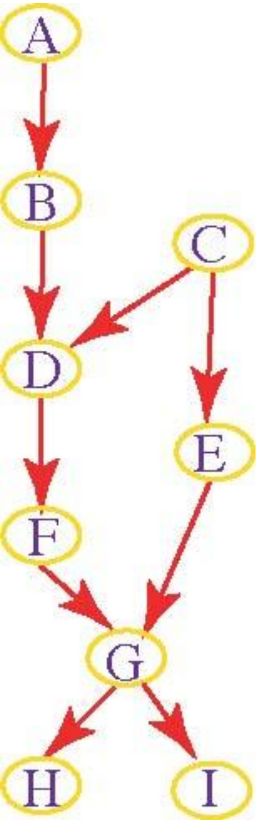
Compute $P(G \mid H=h_1)$.

Previous state:

$$P(G,H) = \sum_{A,B,C,D,E,F,I} f_0(A)\, f_1(B,A)\, f_2(C)\, f_3(D,B,C)\, f_4(E,C)\, f_5(F, D)\, f_6(G,F,E)\, f_9(G)\, f_8(I,G)$$

Elimination ordering A, C, E, I, B, D, F :

$$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B \sum_I f_8(I,G) \underbrace{\sum_E f_6(G,F,E) \underbrace{\sum_C f_2(C)\, f_3(D,B,C)\, f_4(E,C) \underbrace{\sum_A f_0(A)\, f_1(B,A)}_{}}_{}}_{}$$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$
- $f_9(G)$

Variable elimination example(steps 3-4)

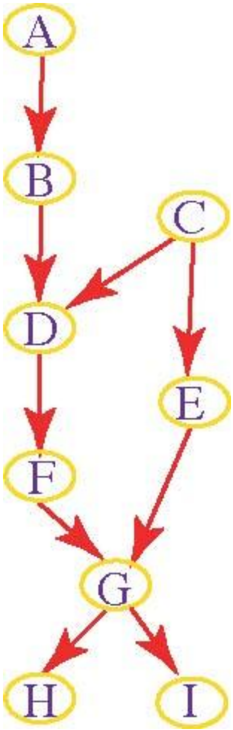
Compute $P(G \mid H=h_1)$. Elimination ordering A, C, E, I, B, D, F.

Previous state:

$$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B \sum_I f_8(I,G) \sum_E f_6(G,F,E) \sum_C f_2(C)\, f_3(D,B,C)\, f_4(E,C) \underline{\sum_A f_0(A)\, f_1(B,A)}$$

Eliminate A:

$$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \underline{\sum_B} f_{10}(B) \sum_I f_8(I,G) \underline{\sum_E} f_6(G,F,E) \underline{\sum_C} f_2(C)\, f_3(D,B,C)\, f_4(E,C)$$



- $\underline{f_0(A)}$
- $\underline{f_1(B,A)}$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

- $f_9(G)$
- $\boxed{f_{10}(B)}$

Variable elimination example(steps 3-4)

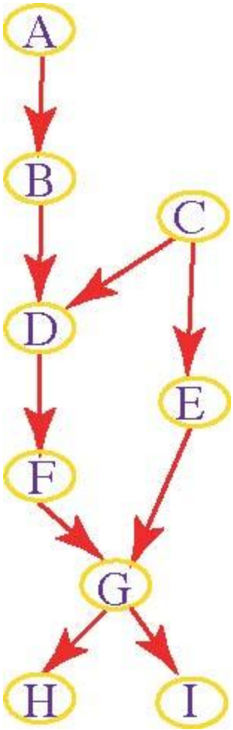
Compute $P(G \mid H=h_1)$. Elimination ordering A, C, E, I, B, D, F.

Previous state:

$$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B) \sum_I f_8(I,G) \sum_E f_6(G,F,E) \underbrace{\sum_C}_{} \underbrace{f_2(C)\, f_3(D,B,C)\, f_4(E,C)}_{}$$

Eliminate C:

$$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B) \sum_I f_8(I,G) \sum_E f_6(G,F,E)\, \underbrace{f_{12}(B,D,E)}_{}$$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$
- $f_9(G)$
- $f_{10}(B)$
- $f_{12}(B,D,E)$

Variable elimination example(steps 3-4)

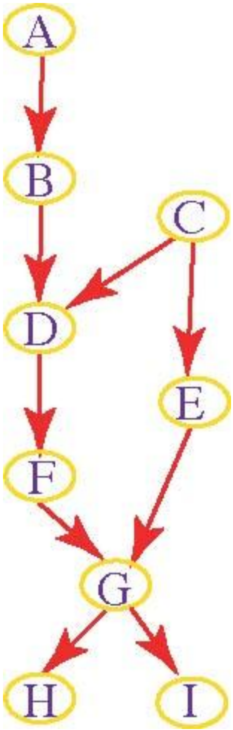
Compute $P(G \mid H=h_1)$. Elimination ordering A, C, E, I, B, D, F.

Previous state:

$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B) \sum_I f_8(I,G) \sum_E f_6(G,F,E) f_{12}(B,D,E)$

Eliminate E:

$P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B)\, f_{13}(B,D,F,G)\, \sum_I f_8(I,G)$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$
- $f_9(G)$
- $f_{10}(B)$
- $f_{12}(B,D,E)$
- $f_{13}(B,D,F,G)$

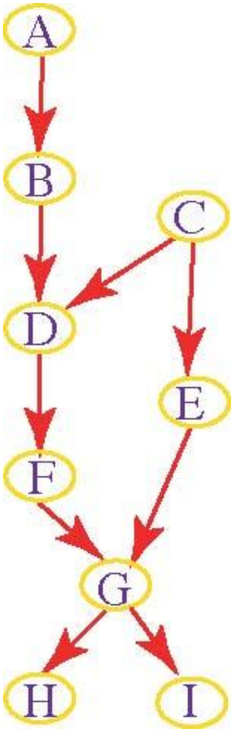
Variable elimination example(steps 3-4)

Compute $P(G \mid H=h_1)$. Elimination ordering A, C, E, I, B, D, F.

Previous state: $P(G,H=h_1) = f_9(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B)\, f_{13}(B,D,F,G)\, \underline{\sum_I f_8(I,G)}$

Eliminate I:

$P(G,H=h_1) = f_9(G)\, \underline{f_{14}(G)} \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B)\, f_{13}(B,D,F,G)$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

- $f_9(G)$
- $f_{10}(B)$
- $f_{12}(B,D,E)$
- $f_{13}(B,D,F,G)$
- $\underline{f_{14}(G)}$

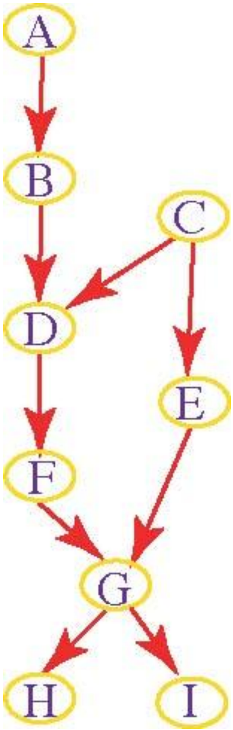
Variable elimination example(steps 3-4)

Compute $P(G \mid H=h_1)$. Elimination ordering A, C, E, I, B, D, F.

Previous state: $P(G,H=h_1) = f_9(G)\, f_{14}(G) \sum_F \sum_D f_5(F, D) \sum_B f_{10}(B)\, f_{13}(B,D,F,G)$

Eliminate B:

$$P(G,H=h_1) = f_9(G)\, f_{14}(G) \sum_F \sum_D f_5(F, D)\, f_{15}(D,F,G)$$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$
- $f_9(G)$
- $f_{10}(B)$
- $f_{12}(B,D,E)$
- $f_{13}(B,D,F,G)$
- $f_{14}(G)$
- $f_{15}(D,F,G)$

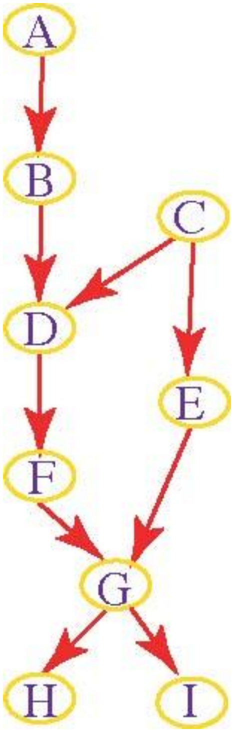
Variable elimination example(steps 3-4)

Compute $P(G \mid H=h_1)$. Elimination ordering A, C, E, I, B, D, F.

Previous state: $P(G,H=h_1) = f_9(G)\, f_{14}(G) \sum_F \sum_D f_5(F, D)\, f_{15}(D,F,G)$

Eliminate D:

$P(G,H=h_1) = f_9(G)\, f_{14}(G) \sum_F f_{16}(F, G)$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

- $f_9(G)$
- $f_{10}(B)$
- $f_{12}(B,D,E)$
- $f_{13}(B,D,F,G)$
- $f_{14}(G)$
- $f_{15}(D,F,G)$
- $f_{16}(F, G)$

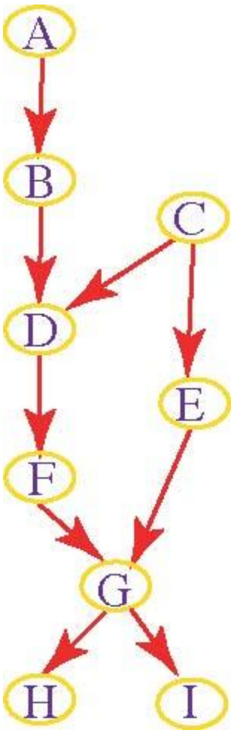
Variable elimination example(steps 3-4)

Compute $P(G \mid H=h_1)$. Elimination ordering A, C, E, I, B, D, F.

Previous state: $P(G,H=h_1) = f_9(G)\, f_{14}(G) \underline{\sum_F f_{16}(F, G)}$

Eliminate F:

$P(G,H=h_1) = \boxed{f_9(G)\, f_{14}(G)\ \underline{f_{17}(G)}}$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$
- $f_9(G)$
- $f_{10}(B)$
- $f_{12}(B,D,E)$
- $f_{13}(B,D,F,G)$
- $f_{14}(G)$
- $f_{15}(D,F,G)$
- $f_{16}(F, G)$
- $f_{17}(G)$

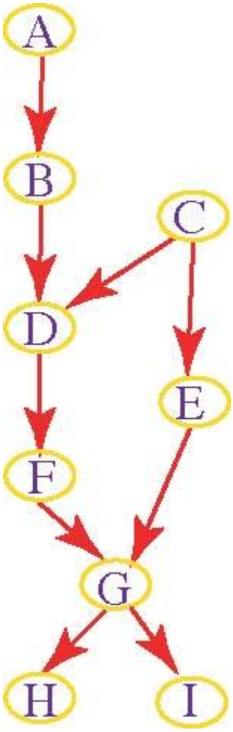
Variable elimination example (step 5)

Compute $\boxed{P(G \mid H=h_1)}$. Elimination ordering A, C, E, I, B, D, F.

Previous state: $P(G,H=h_1) = f_9(G)\ f_{14}(G)\ f_{17}(G)$

Multiply remaining factors:

$$\boxed{P(G,H=h_1)} = f_{18}(G)$$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$
- $f_9(G)$
- $f_{10}(B)$
- $f_{12}(B,D,E)$
- $f_{13}(B,D,F,G)$
- $f_{14}(G)$
- $f_{15}(D,F,G)$
- $f_{16}(F, G)$
- $f_{17}(G)$
- $f_{18}(G)$

Variable elimination example (step 6)

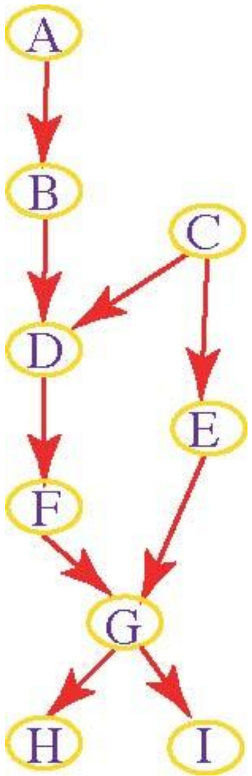
Compute $P(G \mid H=h_1)$. Elimination ordering A, C, E, I, B, D, F.

Previous state:

$P(G,H=h_1) = f_{18}(G)$

Normalize:

$$P(G \mid H=h_1) = f_{18}(G) / \underline{\sum_{g \in dom(G)} f_{18}(G)}$$



- $f_0(A)$
- $f_1(B,A)$
- $f_2(C)$
- $f_3(D,B,C)$
- $f_4(E,C)$
- $f_5(F, D)$
- $f_6(G,F,E)$
- $f_7(H,G)$
- $f_8(I,G)$

- $f_9(G)$
- $f_{10}(B)$
- $f_{12}(B,D,E)$
- $f_{13}(B,D,F,G)$
- $f_{14}(G)$
- $f_{15}(D,F,G)$
- $f_{16}(F, G)$
- $f_{17}(G)$
- $f_{18}(G)$

Lecture Overview

- Recap Intro Variable Elimination
- Variable Elimination
  - Simplifications
  - Example
  - Independence
- Where are we?

Variable elimination and conditional independence

- Variable Elimination looks incredibly painful for large graphs?
- We used conditional independence…..

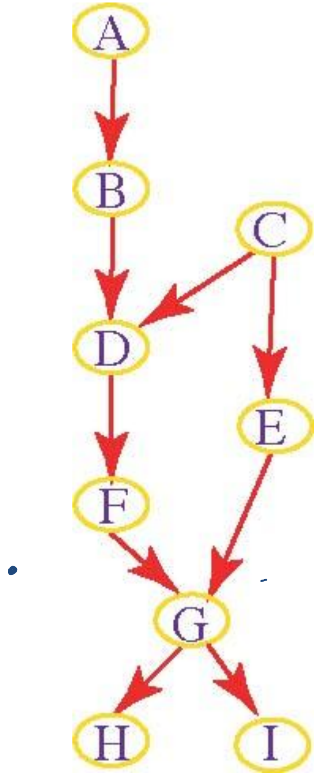
$$P(X_1 \ldots X_n) = \prod P(X_i \mid parents(X_i))$$

  - Can we use it to make variable elimination simpler?

Yes, all the variables from which the query is conditional independent given the observations can be pruned from the Bnet

VE and conditional independence: Example

- All the variables from which the query is conditional independent given the observations can be pruned from the Bnet



e.g., $P(G \mid H=v_1, F= v_2, C=v_3)$.

B, D, E

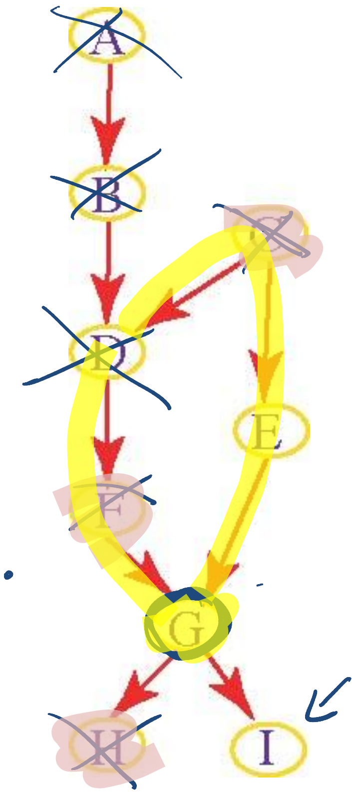
E, D

D, I

B, D, A

VE and conditional independence: Example

- All the variables from which the query is conditional independent given the observations can be pruned from the Bnet



both paths from G to D are blocked

e.g., $P(G \mid H=v_1, F= v_2, C=v_3)$.

G is conditionally independent from A B D given the observed vars H, F, C

Learning Goals for today's class

You can:

- Define **factors**. Derive new factors from existing factors. Apply operations to factors, including assigning, summing out and multiplying factors.
- Carry out **variable elimination** by using factor representation and using the factor operations.
- Use techniques to simplify variable elimination.

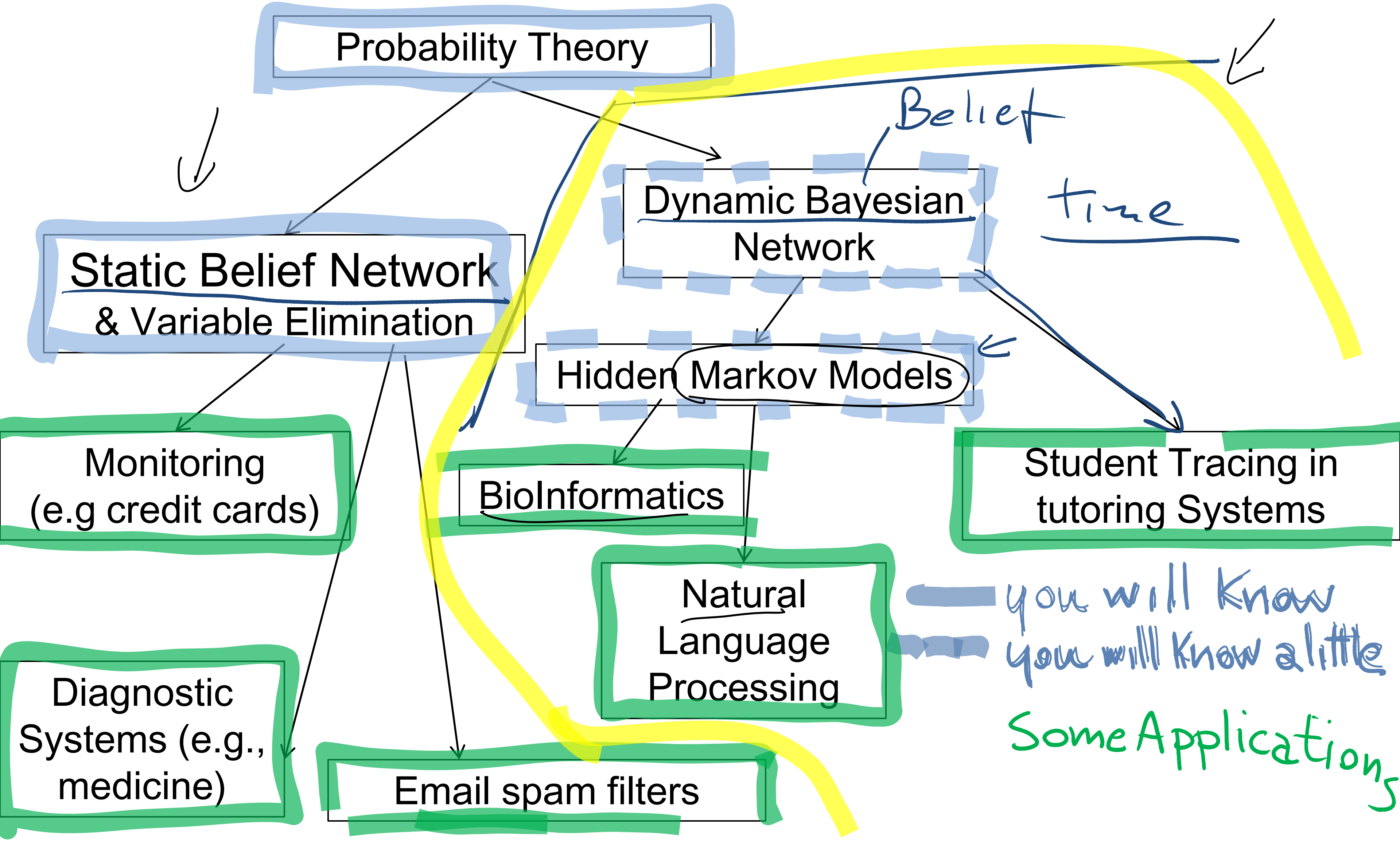
Big Picture: R&R systems

		Environment	
Problem		Deterministic	Stochastic
Static	Constraint Satisfaction	*Vars + Constraints* Arc Consistency Search → for CSP SLS	
	Query	*Logics* CSP Search → for Inference	*Belief Nets* Var. Elimination
Sequential	Planning	*STRIPS* → CSP Search → for complex planning	*Decision Nets* Var. Elimination *Markov Processes* Value Iteration

Representation

Reasoning Technique

Answering Query under Uncertainty



Probability Theory

Static Belief Network & Variable Elimination

Dynamic Bayesian Network

Hidden Markov Models

Monitoring (e.g credit cards)

BioInformatics

Student Tracing in tutoring Systems

Natural Language Processing

Diagnostic Systems (e.g., medicine)

Email spam filters

CPSC 322, Lecture 10

Lecture Overview

- Recap Learning Goals last part of previous lecture
- **Bnets Inference**
  - Intro
  - Factors
  - Variable elimination Algorithm
- **Temporal Probabilistic Models**

Modelling static Environments

So far we have used **Bnets** to perform inference in **static environments**

- For instance, the system keeps collecting evidence to diagnose the cause of a fault in a system (e.g., a car).

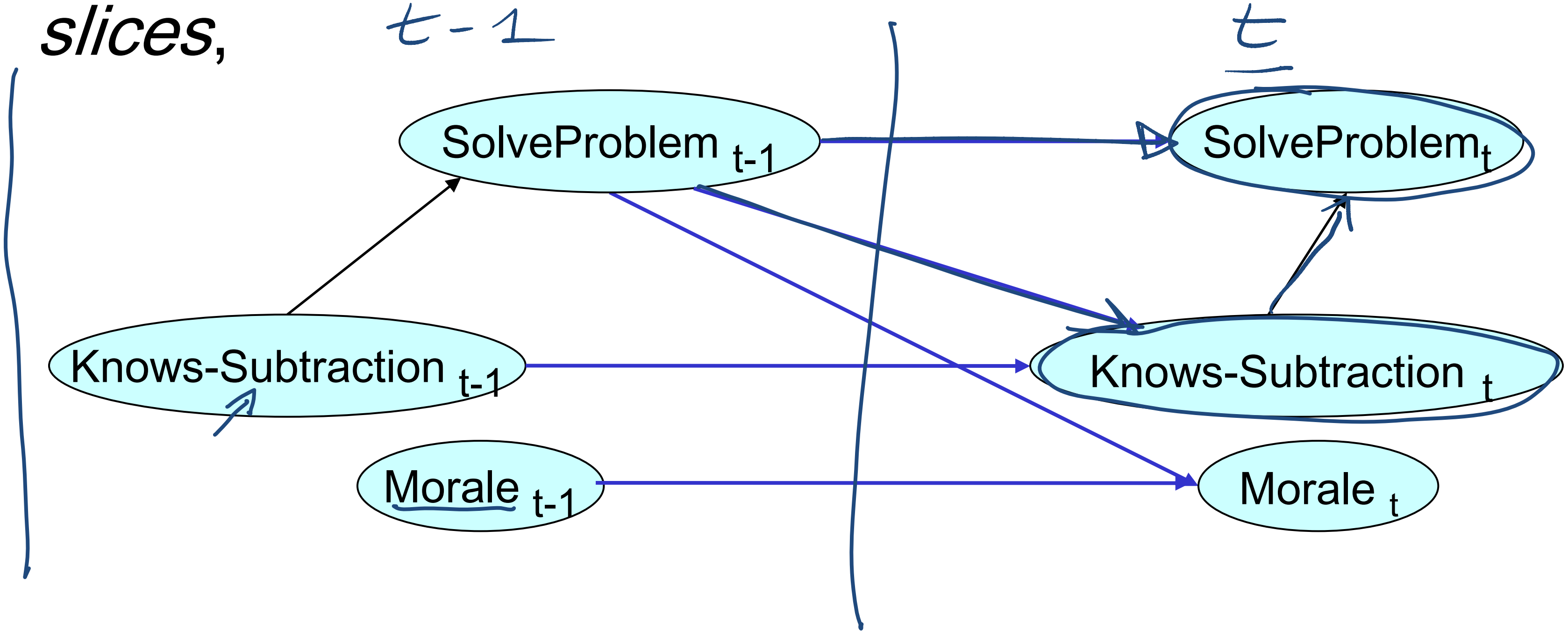
- The environment (values of the evidence, the true cause) does not change as I gather new evidence

- What does change?

The system's beliefs over possible causes

Modeling Evolving Environments

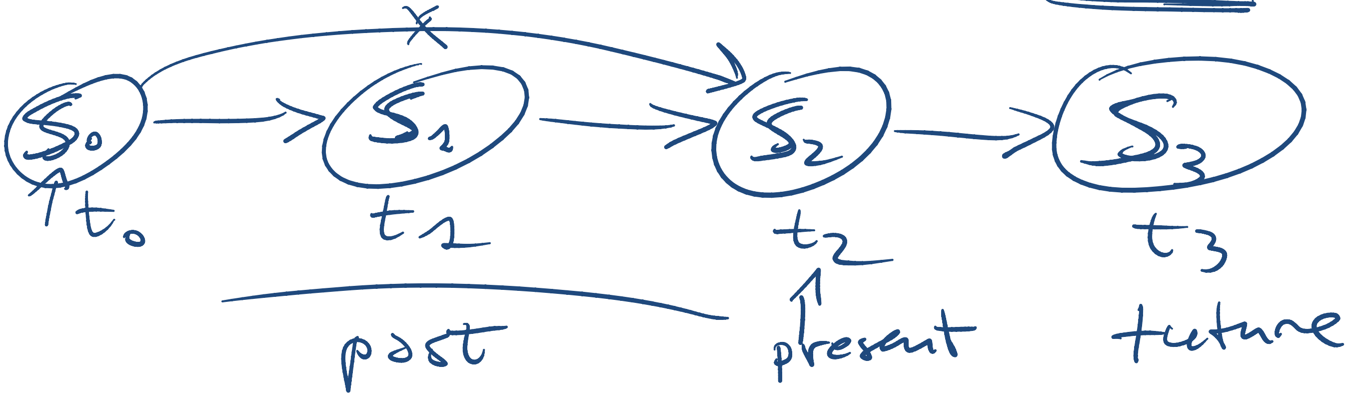
- Often we need to make inferences about **evolving environments.**
- Represent the state of the world at each specific point in time via a series of snapshots, or *time slices*,



Tutoring system tracing student *knowledge* and *morale*

Simplest Possible DBN

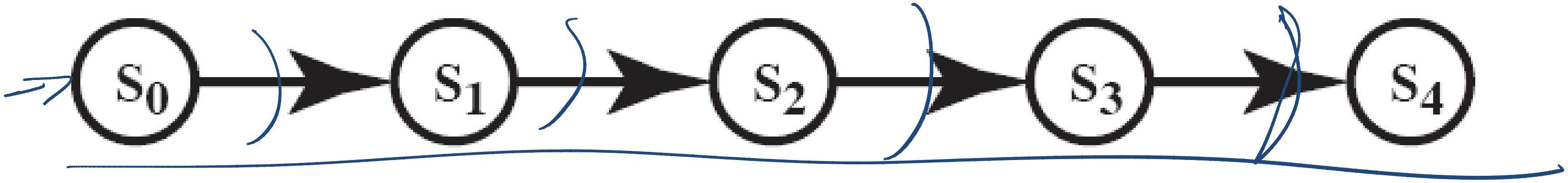
- One random variable for each time slice: let's assume S_t represents the state at time *t*. with domain $\{\boxed{s_1 \ldots s_n}\}$



- Each random variable depends only on the previous one
- Thus $P(S_{t+1} | S_0 \cdots S_t) = P(\underline{S_{t+1}} | \underline{S_t})$
- Intuitively S_t conveys all of the information about the history that can affect the future states.
- "The future is independent of the past given the present."

Simplest Possible DBN (cont')

$P(S_4 | S_3)$



- How many CPTs do we need to specify?

4 $P(S_1 | S_0)$ $P(S_2 | S_1)$ etc.

- *Stationary process assumption:* the mechanism that regulates how state variables change overtime is stationary, that is it can be described by a single transition model
- $P(S_t|S_{t-1})$ is the same for all t

Stationary Markov Chain (SMC)

A stationary Markov Chain : for all t >0

- $\underline{P(S_{t+1}| S_0,\ldots,S_t) = P(S_{t+1}|S_t)}$ and Markov assumption
- $\underline{P(S_{t+1}|S_t)}$ is the same stationary

We only need to specify $P(S_0)$ and $P(S_{t+1} | S_t)$

- Simple Model, easy to specify ←
- Often the natural model ←
- The network can extend indefinitely ←
- Variations of SMC are at the core of most Natural Language Processing (NLP) applications! also used in the PageRank algo (used by Google to rank web pages)

Stationary Markov-Chain: Example

Domain of variable S_i is {t , q, p, a, h, e}

six possible values

We only need to specify…

$P(S_0)$

t	.6
q	.4
p	0
a	0
h	0
e	0

Probability of initial state

Stochastic Transition Matrix

$P(S_{t+1}|S_t)$

Q

S_{t+1}

S_t	t	q	p	a	h	e
t	0	.3	0	.3	.4	0
q	.4	0	.6	0	0	0
p	0	0	1	0	0	0
a	0	0	.4	.6	0	0
h	0	0	0	0	0	1
e	1	0	0	0	0	0

← $P(S_{t+1}|S_t=q)$

← $P(S_{t+1}|S_t=p)$

Markov-Chain: Inference

Probability of a sequence of states $S_0 \ldots S_T$

$$\underline{P(S_0,\ldots,S_T)} = P(S_0)\,P(S_1|S_0)\,P(S_2|S_1) \ldots\ldots$$

$$S_0 \rightarrow S_1 \rightarrow S_2$$

$$P(u,e,e) \rightarrow$$

$P(S_0)$

t	.6
q	.4
p	0
a	0
h	0
e	0

$P(S_{t+1}|S_t)$

	t	q	p	a	h	e
t	0	.3	0	.3	.4	0
q	.4	0	.6	0	0	0
p	0	0	1	0	0	0
a	0	0	.4	.6	0	0
h	0	0	0	0	0	1
e	1	0	0	0	0	0

Example:

$$\underline{P(t, q, p)} =$$

$$P(t) * P(q|t) * P(p|q)$$

$$.6 * .3 * .6 = .108$$

Next Class

Decision Networks *(TextBook 9.2-9.3)*

Course Elements

- Assignment 4 has been posted
- Will post questions to prepare for final